Open Security Research

Sponsored by Foundstone

Tuesday, September 16, 2014

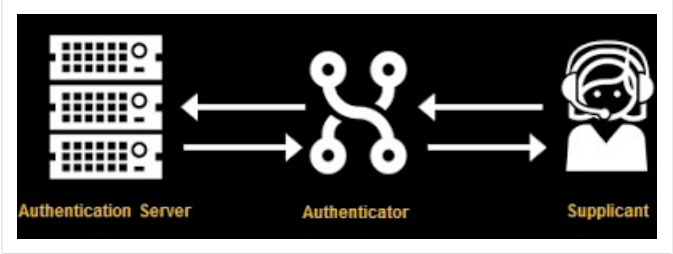
hostapd-wpe: Now with More Pwnage!

By Brad Antoniewicz.

A major component of hacking IEEE 802.11 wireless networks is targeting the client's system. This is because of the trusting nature of wireless and corporate systems can be tricky to configure correctly. But don't forget that the same client-side attacks against 802.11 wireless networks can be used on wired networks with port security when the attacker has physical access to a workstation or access switch. hostapd-wpe provides a means to execute client side attacks on wired and wireless networks, and in this blog post we'll cover hostapd-wpe's latest features.

Background

Both IEEE 802.11 and Ethernet can utilize a security standard called IEEE 802.1x that provides the opportunity for "the network" to authenticate the connecting user. In wireless networks, this is part of WPA Enterprise. 802.1x relies heavily on the Extensible Authentication Protocol (EAP) to send messages between the connecting user (supplicant) and the authentication server. To be as flexible as possible, there are different "EAP Types" which offer different authentication options chosen by the network administrator. For instance, PEAP first sets up a TLS tunnel between the client and server, then sends a username and password within that tunnel.



An opportunity to attack networks running 802.1x exists if the attacker can position themselves between the client and the authentication server. If that happens and the user is configured to blindly trust the network they're connecting to, the user may naively trust an impostor authentication server set up by the attacker and send its username and password to it.

This attack was first implemented in a tool myself and Josh Wright wrote called FreeRADIUS-WPE and recently implemented in hostapd-wpe.

Features

FreeRADIUS-WPE is a great approach to performing client-side attacks against 802.1x/EAP but since its only an authentication server, you still need to create an authenticator. The authenticator of choice most commonly hostapd because it can be run in software, is generally easy to set up, and supports wired and wireless attacks. There's one thing about hostapd that I didn't mention: it can also be an authentication server! So we can move the "WPE functionality" from FreeRADIUS-WPE to hostapd and we eliminate an unneeded layer of complexity!

Our Regular Authors

Brad Antoniewicz
Tony Lee
Gursev Singh Kalra
Robert Portvliet
Melissa Augustine
Paul Ambrosini
Tushar Dalvi

Popular Posts

Getting Started with GNU Radio and RTL-SDR (on Backtrack)
Deconstructing a Credit Card's Data
Using Mimikatz to Dump Passwords!
Windows DLL Injection Basics
Comcast and DOCSIS 3.0 - Worth the upgrade?
Identifying Malware Traffic with Bro and the Collective Intelligence Framework (CIF)
Deobfuscating Potentially Malicious URLs - Part 1
Top 10 Oracle Steps to a Secure Oracle Database Server
Hacking KeyLoggers
Setting up a Password Cracking Server

Blog Archive

▼ 2014 (27)
► November (1)
▼ September (2)
hostapd-wpe: Now with More Pwnage!
Face Smack: A CSAW CTF Challenge

Impersonation Attacks

hostapd-wpe's core feature is authentication server/authenticator impersonation. It simply logs authentication data from the client to a file and outputs it to the screen. It currently supports the following EAP Types:

- EAP-FAST/MSCHAPv2 (Phase 0)
- PEAP/MSCHAPv2
- EAP-TTLS/MSCHAPv2
- EAP-TTLS/MSCHAP
- EAP-TTLS/CHAP
- EAP-TTLS/PAP

Logging

Data from the client relevant to the attack is stored in `hostapd-wpe.log` within the directory where hostapd-wpe was called from. This could be credentials are heartbleed data.

The log file location can be configured within the `hostapd-wpe.conf` configuration file by the `wpe_logfile` option.

Credential Format

For EAP Types that utilize MSCHAPv2, hostapd-wpe outputs the challenge and response in the standard WPE format and john's NETNTLM format.

This feature is enabled by default.

Requests for Less Secure Types

hostapd-wpe is configured by default to request cleartext and other less-secure EAP-Types (e.g. PAP) from the client. In some cases a client maybe configured to support multiple EAP-Types, so this acts as sort of a "downgrade" attack.

This feature is enabled by default through the `hostapd-wpe.eap_user`.

Return EAP-Success

At the end of a successful authentication, the Authentication Server sends an EAP-Success message. hostapd-wpe will always return an EAP-Success so that the client believes they are successfully authenticated and continues normal connection procedures. Assuming the attacker provides the appropriate requirements to establish a connection (IP, DNS, etc..) - the attacker can leverage this to MiTM client traffic or otherwise attack the client.

This feature can be invoked using the `-s` option via the command line.

Cupid (Heartbleed) Client Attacks

hostapd-wpe implements Cupid or Heartbleed attacks against connecting clients.

This feature can be invoked using the `-c` option via the command line. The following configuration options exist within the `hostapd-wpe.conf` configuration file, however default settings are recommended:

```
wpe_hb_send_before_handshake=0    # Heartbleed True/False (Default: 1)
wpe_hb_send_before_appdata=0      # Heartbleed True/False (Default: 0)
wpe_hb_send_after_appdata=0       # Heartbleed True/False (Default: 0)
wpe_hb_payload_size=0             # Heartbleed 0-65535 (Default: 50000)
wpe_hb_num_repeats=0              # Heartbleed 0-65535 (Default: 1)
wpe_hb_num_tries=0                # Heartbleed 0-65535 (Default: 1)
```


- ► August (2)
- ► July (2)
- ► June (4)
- ► May (2)
- ► April (4)
- ► March (3)
- ► February (3)
- ► January (4)

► 2013 (40)

► 2012 (60)

► 2011 (15)

Karma-Style Probe Responses

Some 802.11 clients send out probe requests to determine if the wireless network they're configured for is nearby. When Karma-Style Probe Responses are enabled, hostapd-wpe will look for client probe requests and immediately change the SSID it is broadcasting to match the probe request of the client.

This feature can be invoked using the `-k` option via the command line.

Installation

Homepage: https://github.com/OpenSecurityResearch/hostapd-wpe

To get hostapd-wpe running on Kali or whatever other Debian based system you first have to install its dependencies:

```
apt-get update
apt-get install libssl-dev libnl-dev
```

hostapd-wpe is patch to hostapd, so you'll next have to download the hostapd source and apply the patch:

```
git clone https://github.com/OpenSecurityResearch/hostapd-wpe
wget http://hostap.epitest.fi/releases/hostapd-2.2.tar.gz
tar -zxf hostapd-2.2.tar.gz
cd hostapd-2.2
patch -p1 < ../hostapd-wpe/hostapd-wpe.patch
```

Now you can build

```
cd hostapd
make
```

You'll also need some certificates set up, you can do this with the bootstrap script:

```
cd ../../hostapd-wpe/certs
./bootstrap
```

Look at `hostapd-wpe.conf` and set the `interface` and `driver` values accordingly to your needs (and perhaps the `ssid`, `hw_mode`, and `channel` for 802.11). Then to run:

```
cd ../../hostapd-2.2/hostapd
sudo ./hostapd-wpe hostapd-wpe.conf
```

Enjoy!

Posted by OpenSecurity Research at 2:35 PM

+1 +4 Recommend this on Google

Labels: network penetration testing, penetration testing, wireless, wireless hacking

2 comments:

Anonymous September 16, 2014 at 3:07 PM

Very good work ! Thx !

Reply

Anonymous December 2, 2014 at 11:02 AM

Thanks for the good work! However it doesn't work when capturing MSCHAPv2 hashes from Windows clients due to their implementation. Freeradius-wpe had a nice hack with the "with ntlm_hack" option which is missing. Do you plan on adding this hack to hostapd-wpe too ?

Reply

Enter your comment...

Comment as: Google Account

Publish Preview

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)